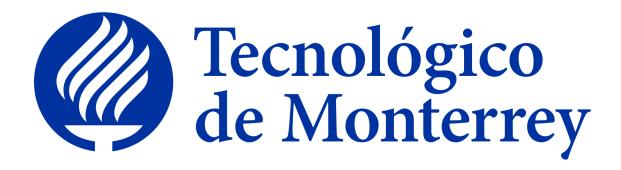
Instituto Tecnológico y de Estudios Superiores de Monterrey Campus Monterrey



Implementación de robótica inteligente TE3002B, Grupo 501

Canny Edge

Elaborado Por:

| Emilio Rizo De la Mora | l A01721612 |
|---------------------------------|-------------|
| Jose Pablo Cedano Serna | l A00832019 |
| Luis Antonio Zermeño De Gorordo | l A01781835 |
| Omar Flores Sanchéz | l A01383530 |
| Luis Mario Lozoya Chairez | l A00833364 |
| Jorge Axel Castruita Bretado | I A00832843 |
| Rodrigo Escandón López Guerrero | l A01704287 |

Dr. Luis Alberto Muñoz

1. Introducción

La detección de bordes en imágenes y videos es una técnica fundamental en el procesamiento de imágenes que permite identificar las regiones donde hay un cambio significativo en la intensidad de los pixeles. Uno de los algoritmos más utilizados para esta tarea es el algoritmo de Canny Edge Detection, desarrollado por John F. Canny en 1986. Este algoritmo es ampliamente utilizado en diversas aplicaciones, como visión por computadora, reconocimiento de objetos y seguimiento de objetos en movimiento.

2. Descripción

El proyecto desarrollado consiste en la implementación de un sistema de detección de bordes en un video utilizando el algoritmo de Canny Edge Detection. El sistema se encarga de leer un archivo de video, mostrar cada fotograma en una ventana y aplicar el algoritmo de detección de bordes para resaltar las transiciones más significativas de intensidad de color en cada fotograma.

3. Características

- Lectura de Video: El programa es capaz de leer un archivo de video especificado y cargarlo.
- Detección de Bordes: Utiliza el algoritmo de Canny Edge Detection para detectar los bordes en cada fotograma del video, resaltando las regiones de cambio brusco en la intensidad de color.
- Visualización de Resultados: Muestra dos ventanas: una con el video original y otra con el resultado de la detección de bordes..

4. Código

```
import cv2

# Read the video file
video_capture = cv2.VideoCapture(r'messi.mp4')

# Check if the video file opened successfully
if not video_capture.isOpened():
    print("Error: Unable to open video file.")
    exit()

while True:
    # Read a frame from the video
    ret, frame = video_capture.read()

# Check if the frame was read successfully
if not ret:
    # If the video ends, rewind it to the beginning
```

```
video_capture.set(cv2.CAP_PROP_POS_FRAMES, 0)
    cv2.imshow('Video', frame)
    frame_gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    frame blur = cv2.GaussianBlur(frame_gray, (3, 3), 0)
    edges = cv2.Canny(image=frame_blur, threshold1=100, threshold2=200)
   cv2.imshow('Canny Edge Detection', edges)
   if cv2.waitKey(25) \& 0xFF == ord(' '):
video capture.release()
cv2.destroyAllWindows()
```

5. Resultados

Video de funcionalidad:

canny_edge.mp4

Repositorio de github:

https://github.com/jpcedano/canny-edge